Results of the 2013 Termination Competition

René Thiemann

Computational Logic
Institute of Computer Science
University of Innsbruck

Rewriting Techniques and Applications, June 26, 2013
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time
- Bugs and Problems
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time
- Bugs and Problems
The competition in numbers

- 1143 problems selected out of 6943 in TPDB
The competition in numbers

- 1143 problems selected out of 6943 in TPDB
- 10 different provers, 1 certifier, 1 execution platform, 3 servers
The competition in numbers

- 1143 problems selected out of 6943 in TPDB
- 10 different provers, 1 certifier, 1 execution platform, 3 servers
- 12 categories
The competition in numbers

- 1143 problems selected out of 6943 in TPDB
- 10 different provers, 1 certifier, 1 execution platform, 3 servers
- 12 categories
- 4903 results
  - 2025 YES, termination is proven (sometimes with complexity)
  - 295 NO, termination does not hold
  - 2102 MAYBE
  - 481 TIMEOUT
- 764 ACCEPT, (non)termination or complexity proof is correct
- 12 REJECT, (non)termination or complexity proof is buggy
- 38 hours of computation
The competition in numbers

- 1143 problems selected out of 6943 in TPDB
- 10 different provers, 1 certifier, 1 execution platform, 3 servers
- 12 categories
- 4903 results + 776 certification results
  - 2025 YES, termination is proven (sometimes with complexity)
  - 295 NO, termination does not hold
  - 2102 MAYBE
  - 481 TIMEOUT
  - 764 ACCEPT, (non)termination or complexity proof is correct
  - 12 REJECT, (non)termination or complexity proof is buggy
- 38 hours of computation
The competition in numbers

- 1143 problems selected out of 6943 in TPDB
- 10 different provers, 1 certifier, 1 execution platform, 3 servers
- 12 categories
- 4903 results + 776 certification results
  - 2025 YES, termination is proven (sometimes with complexity)
  - 295 NO, termination does not hold
  - 2102 MAYBE
  - 481 TIMEOUT
  - 764 ACCEPT, (non)termination or complexity proof is correct
  - 12 REJECT, (non)termination or complexity proof is buggy
- 38 hours of computation
The setup

- every prover has 1 minute to solve each termination / complexity problem
The setup

- every prover has 1 minute to solve each termination / complexity problem
- every job has exclusive access to server
  - 64 GB RAM
  - 16 cores (8 Dual-Core Opteron\textsuperscript{tm} 8220 at 2.8 GHz)
  - 64-bit linux
The setup

- every prover has 1 minute to solve each termination / complexity problem
- every job has exclusive access to server
  - 64 GB RAM
  - 16 cores (8 Dual-Core Opteron\textsuperscript{tm} 8220 at 2.8 GHz)
  - 64-bit linux
- organization of TPDB
  - submissions are grouped into families
  - duplicates are eliminated
    - modulo symbol-names, variable-names, order of rules
    - currently only for SRS / TRS
  \[\Rightarrow\] extended $\alpha$-equivalence of Sabel et. al, GI-hard
The setup

- every prover has 1 minute to solve each termination / complexity problem
- every job has exclusive access to server
  - 64 GB RAM
  - 16 cores (8 Dual-Core Opteron™ 8220 at 2.8 GHz)
  - 64-bit linux
- organization of TPDB
  - submissions are grouped into families
  - duplicates are eliminated
    - modulo symbol-names, variable-names, order of rules
    - currently only for SRS / TRS
  \[\implies \text{extended } \alpha\text{-equivalence of Sabel et. al, GI-hard}\]
- random selection of subset of problems of TPDB such that
  - every family of problems is taken into account
  - everything can be computed during RTA
This year: real competition

- new rule: only consider categories with ≥ 2 participants
  ⇒ several categories have been dropped this year
    - complexity: TCT
      - derivational innermost
    - termination: AProVE
      - Haskell
      - Logic programming with cut
      - Prolog
      - conditional / innermost / outermost / context-sensitive TRS
    - certification: CeTA
      - complexity (derivational and runtime, full and innermost)
      - termination (conditional / innermost / outermost / relative / standard TRS)
    - CeTA was still invoked to check whether competing tools with certifiable output produced correct proofs
This year: real competition

- new rule: only consider categories with \( \geq 2 \) participants
- several categories have been dropped this year
  - complexity: TCT
    - derivational innermost
  - termination: AProVE
    - Haskell
    - Logic programming with cut
    - Prolog
    - conditional / innermost / outermost / context-sensitive TRS
- certification: CeTA
  - complexity (derivational and runtime, full and innermost)
  - termination (conditional / innermost / outermost / relative / standard TRS)
- CeTA was still invoked to check whether competing tools with certifiable output produced correct proofs
- Hint: surprise tools in these categories within next competition
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time
- Bugs and Problems
Competitors

- AProVE, RWTH Aachen University (Giesl et. al)
  - also certified variant AProVE-CeTA
- muTerm, Universitat Politècnica de València (Lucas et. al)
- Thor, Universitat Politècnica de Catalunya (Rubio)
- TTT2, University of Innsbruck (Middeldorp et. al)
  - also certified variant TTT2-Cert
- Wanda, VU University Amsterdam / University of Innsbruck (Kop)
### Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>AProVE</td>
<td>muTerm</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>HO rewriting</td>
<td>HOT</td>
<td>Wanda</td>
<td>Thor</td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
## Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE (77 %)</td>
<td>TTT2 (71)</td>
<td>AProVE-CeTA (63)</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>AProVE</td>
<td>muTerm</td>
<td>Thor</td>
</tr>
<tr>
<td>HO rewriting</td>
<td>HOT</td>
<td>Wanda</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
## Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE (77 %)</td>
<td>TTT2 (71)</td>
<td>AProVE-CeTA (63)</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE (49 %)</td>
<td>AProVE-CeTA (38)</td>
<td>TTT2 (34)</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE</td>
<td>AProVE-CeTA</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>AProVE</td>
<td>muTerm</td>
<td>Thor</td>
</tr>
<tr>
<td>HO rewriting</td>
<td>HOT</td>
<td>Wanda</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
## Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE (77 %)</td>
<td>TTT2 (71)</td>
<td>AProVE-CeTA (63)</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE (49 %)</td>
<td>AProVE-CeTA (38)</td>
<td>TTT2 (34)</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE (86 %)</td>
<td>AProVE-CeTA (72)</td>
<td>TTT2-Cert</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE</td>
<td>muTerm</td>
<td>Thor</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>HOT</td>
<td>Wanda</td>
<td></td>
</tr>
<tr>
<td>HO rewriting</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
## Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE (77 %)</td>
<td>TTT2 (71)</td>
<td>AProVE-CeTA (63)</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE (49 %)</td>
<td>AProVE-CeTA (38)</td>
<td>TTT2 (34)</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE (86 %)</td>
<td>AProVE-CeTA (72)</td>
<td>TTT2 (65)</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE (66 %)</td>
<td>AProVE-CeTA (61)</td>
<td>TTT2(-Cert) (44)</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>AProVE</td>
<td>muTerm</td>
<td>Thor</td>
</tr>
<tr>
<td>HO rewriting</td>
<td>HOT</td>
<td>Wanda</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
## Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE (77 %)</td>
<td>TTT2 (71)</td>
<td>AProVE-CeTA (63)</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE (49 %)</td>
<td>AProVE-CeTA (38)</td>
<td>TTT2 (34)</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE (86 %)</td>
<td>AProVE-CeTA (72)</td>
<td>TTT2 (65)</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE (66 %)</td>
<td>AProVE-CeTA (61)</td>
<td>TTT2(-Cert) (44)</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>AProVE (81 %)</td>
<td>muTerm (74)</td>
<td>Thor</td>
</tr>
<tr>
<td>HO rewriting</td>
<td>HOT</td>
<td>Wanda</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
## Term rewriting variants

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>string rewriting</td>
<td>AProVE (77 %)</td>
<td>TTT2 (71)</td>
<td>AProVE-CeTA (63)</td>
</tr>
<tr>
<td>strings (relative)</td>
<td>AProVE (49 %)</td>
<td>AProVE-CeTA (38)</td>
<td>TTT2 (34)</td>
</tr>
<tr>
<td>term rewriting</td>
<td>AProVE (86 %)</td>
<td>AProVE-CeTA (72)</td>
<td>TTT2 (65)</td>
</tr>
<tr>
<td>terms (relative)</td>
<td>AProVE (66 %)</td>
<td>AProVE-CeTA (61)</td>
<td>TTT2(-Cert) (44)</td>
</tr>
<tr>
<td>terms (equational)</td>
<td>AProVE (81 %)</td>
<td>muTerm (74)</td>
<td></td>
</tr>
<tr>
<td>HO rewriting</td>
<td>Wanda (82 %)</td>
<td>Thor (63)</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, AProVE-CeTA, TTT2, TTT2-Cert, Wanda
- new examples for higher-order rewriting
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time
- Bugs and Problems
Competitors

- AProVE, RWTH Aachen University (Giesl et. al)
- Julia, University of Verona and University of Reunion (Mesnard et. al)
- polytool, KU Leuven (Nguyen et. al)
- pTNT, KU Leuven (Voets and De Schreye)
Rewriting against dedicated tools . . .

- AProVE: transformation to rewriting
- Julia, polytool, pTNT: tools outside rewriting community

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>logic programming</td>
<td>AProVE</td>
<td>polytool</td>
<td>pTNT</td>
</tr>
<tr>
<td>Java bytecode</td>
<td>AProVE</td>
<td>Julia</td>
<td></td>
</tr>
<tr>
<td>Java bytecode rec.</td>
<td>AProVE</td>
<td>Julia</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, Julia
- new examples for Java categories
**Rewriting against dedicated tools**

- **AProVE**: transformation to rewriting
- **Julia, polytool, pTNT**: tools outside rewriting community

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>logic programming</td>
<td>AProVE (74 %)</td>
<td>polytool (70)</td>
<td>pTNT (22)</td>
</tr>
<tr>
<td>Java bytecode</td>
<td>AProVE</td>
<td>Julia</td>
<td></td>
</tr>
<tr>
<td>Java bytecode rec.</td>
<td>AProVE</td>
<td>Julia</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, Julia
- new examples for Java categories
Rewriting against dedicated tools . . .

- AProVE: transformation to rewriting
- Julia, polytool, pTNT: tools outside rewriting community

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>logic programming</td>
<td>AProVE (74 %)</td>
<td>polytool (70)</td>
<td>pTNT (22)</td>
</tr>
<tr>
<td>Java bytecode</td>
<td>AProVE (91 %)</td>
<td>Julia (38)</td>
<td></td>
</tr>
<tr>
<td>Java bytecode rec.</td>
<td>AProVE</td>
<td>Julia</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, Julia
- new examples for Java categories
Rewriting against dedicated tools . . .

- **AProve**: transformation to rewriting
- **Julia, polytool, pTNT**: tools outside rewriting community

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>logic programming</td>
<td>AProVE (74 %)</td>
<td>polytool (70)</td>
<td>pTNT (22)</td>
</tr>
<tr>
<td>Java bytecode</td>
<td>AProVE (91 %)</td>
<td>Julia (38)</td>
<td></td>
</tr>
<tr>
<td>Java bytecode rec.</td>
<td>AProVE (88 %)</td>
<td>Julia (37)</td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, Julia
- new examples for Java categories
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time
- Bugs and Problems
Competition analysis

Competitors

- AProVE: RWTH Aachen University (Giesl et. al)
- CaT: University of Innsbruck (Korp, Zankl)
  - also certified variant CaT-Cert
- TCT: University of Innsbruck (Avanzini, Moser, Schnabl)
  - also certified variant TCT-certify
### Complexity results

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>derivational complexity</td>
<td>CaT</td>
<td>TCT</td>
<td>CaT-Cert</td>
</tr>
<tr>
<td>runtime complexity</td>
<td>TCT</td>
<td>CaT</td>
<td>TCT-certify</td>
</tr>
<tr>
<td>runtime innermost</td>
<td>AProVE</td>
<td>TCT</td>
<td>TCT-certify</td>
</tr>
</tbody>
</table>

- new implementations: AProVE, TCT
### Complexity results

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>derivational complexity</td>
<td>CaT (299 points)</td>
<td>TCT (282)</td>
<td>TCT-certify (197)</td>
</tr>
<tr>
<td>runtime complexity</td>
<td>TCT</td>
<td>CaT</td>
<td>TCT-certify</td>
</tr>
<tr>
<td>runtime innermost</td>
<td>AProVE</td>
<td>TCT</td>
<td>TCT-certify</td>
</tr>
</tbody>
</table>

- new implementations: AProVE, TCT
### Complexity results

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>derivational complexity</td>
<td>CaT (299 points)</td>
<td>TCT (282)</td>
<td>TCT-certify (197)</td>
</tr>
<tr>
<td>runtime complexity</td>
<td>TCT (180)</td>
<td>CaT (104)</td>
<td>TCT-certify (60)</td>
</tr>
<tr>
<td>runtime innermost</td>
<td>AProVE</td>
<td>TCT</td>
<td>TCT-certify</td>
</tr>
</tbody>
</table>

- new implementations: AProVE, TCT
### Complexity results

<table>
<thead>
<tr>
<th>Category</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>derivational complexity</td>
<td>CaT (299 points)</td>
<td>TCT (282)</td>
<td>TCT-certify (197)</td>
</tr>
<tr>
<td></td>
<td>TCT (180)</td>
<td>CaT (104)</td>
<td>TCT-certify (60)</td>
</tr>
<tr>
<td>runtime complexity</td>
<td>AProVE (295)</td>
<td>TCT (279)</td>
<td>TCT-certify (60)</td>
</tr>
<tr>
<td>runtime innermost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- new implementations: AProVE, TCT
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time
- Bugs and Problems
### Fastest tools

<table>
<thead>
<tr>
<th>Average time</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>Thor</td>
</tr>
<tr>
<td>1.5</td>
<td>Wanda</td>
</tr>
<tr>
<td>6.4</td>
<td>polytool</td>
</tr>
<tr>
<td>17.4</td>
<td>TTT2</td>
</tr>
<tr>
<td>18.9</td>
<td>AProVE</td>
</tr>
<tr>
<td>19.0</td>
<td>CaT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Successes per minute</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>178.7</td>
<td>Thor</td>
</tr>
<tr>
<td>32.0</td>
<td>Wanda</td>
</tr>
<tr>
<td>6.5</td>
<td>polytool</td>
</tr>
<tr>
<td>2.3</td>
<td>muTerm</td>
</tr>
<tr>
<td>2.2</td>
<td>AProVE</td>
</tr>
<tr>
<td>2.2</td>
<td>TTT2</td>
</tr>
</tbody>
</table>
Overview

- Introduction
- Termination of term rewriting
- Termination of programs
- Complexity analysis
- Execution time

- Bugs and Problems
• Problem: Network-failure between competition servers
  • several correct proofs have been rejected
    (FAILED VALIDATION)
  ⇒ already resolved by rerunning these experiments
Bugs and Problems

- **Problem**: Network-failure between competition servers
  - several correct proofs have been rejected (FAILED VALIDATION)
  ⇒ already resolved by rerunning these experiments

- **Bug**: TTT2-certify
  - typo in code led to removal of too many rules
  ⇒ 1 YES-NO conflict, 12 buggy proofs detected via certifier
  - no impact on TTT2, since there other strategy is used
  ⇒ rerun with fixed version soonish
Summary

- interesting competition (subjective reason: bug detected)
Summary

- interesting competition (subjective reason: bug detected)
- boring competition
  - few changes in relative power of tools
  - few new examples
  - no new participants

⇒ new examples and new participants are welcome
Summary

- interesting competition (subjective reason: bug detected)
- boring competition
  - few changes in relative power of tools
  - few new examples
  - no new participants

⇒ new examples and new participants are welcome

- planned later this year: full run on all examples

⇒ thanks
- all participants to
  - Femke for the slot to present the results
  - Hans for organizing a convenient display during the conference

René Thiemann (University of Innsbruck)  Results of the 2013 Termination Competition
Summary

- interesting competition (subjective reason: bug detected)
- boring competition
  - few changes in relative power of tools
  - few new examples
  - no new participants

⇒ new examples and new participants are welcome
- planned later this year: full run on all examples
- thanks
  - all participants to
  - Femke for the slot to present the results
  - Hans for organizing a convenient display during the conference